

RESEARCH ARTICLE

Efficient Facial Landmark Detection via Prior Knowledge-Guided Agents

SEONGMIN JIN¹ AND DOO SEOK JEONG^{1,2}, (Member, IEEE)¹Division of Materials Science and Engineering, Hanyang University, Seoul 04763, Republic of Korea²Department of Semiconductor Engineering, Hanyang University, Seoul 04763, Republic of Korea

Corresponding author: Doo Seok Jeong (dooseokj@hanyang.ac.kr)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korean Government [Ministry of Science and ICT (MSIT)] under Grant RS-2025-02218733 and Grant RS-2023-00253914.

ABSTRACT We present a highly efficient, agent-based framework for facial landmark detection that prioritizes model compactness and computational efficiency over maximum accuracy. Unlike conventional approaches that rely on large, fully supervised models, our method assigns each agent to a specific landmark, enabling it to infer its position solely from local observations and prior knowledge without explicit location awareness or inter-agent communication. Prior knowledge is modeled in two embedding spaces—feature and coordinate—using class-conditional Gaussian distributions. Agents determine landmark detection confidence from deviations to these priors and estimate adaptive hopping behavior through a lightweight policy network. A multi-stage detection strategy further reduces redundant computation by detecting sub-landmarks relative to core landmarks. While our method produces slightly higher normalized mean error than state-of-the-art (SoTA) methods, it achieves over $16\times$ and $41\times$ improvements in space and time complexities, respectively.

INDEX TERMS Facial landmark detection, knowledge modeling, prior knowledge.

I. INTRODUCTION

Facial landmark detection is a fundamental component in many computer vision applications, including face recognition, expression analysis, and 3D face reconstruction [1], [2], [3]. Over the past decade, advances in deep learning have greatly improved detection accuracy. However, the increasing demand for deployment on resource-constrained platforms, such as mobile devices and AR/VR headsets, has shifted attention toward efficiency-oriented solutions. Such platforms impose strict limits on power consumption, computation, and memory, motivating the need for new algorithms that balance accuracy with efficiency.

Facial landmark detection aims to localize key facial points in 2D images. State-of-the-art (SoTA) methods often achieve high accuracy using large-scale models, especially those based on supervised heatmap and coordinate regression [4], [5], [6], [7], [8]. However, these approaches require

substantial computational and memory resources. In contrast, our method adopts a lightweight, agent-based approach that leverages prior knowledge and local observations to detect landmarks efficiently. To support efficient landmark search, we exploit a spatio-semantic latent space [9] that jointly preserves object identity and spatial distance, enabling agents to determine landmark detection confidence and adaptive hopping behavior. Although our normalized mean error (NME) is higher than that of SoTA models, our approach offers orders-of-magnitude gains in efficiency with only 577k parameters and less than 30 MFLOPs, compared to 9.7–67M parameters and 1.2–26.8 GFLOPs for conventional methods. Furthermore, experiments with more expressive prior knowledge models indicate that the proposed framework has the potential to achieve strong localization accuracy while preserving its computational efficiency. Our contributions are as follows:

- We propose an agent-based framework that localizes facial landmarks using only local observations and prior knowledge without relying on absolute coordinate information, or requiring inter-agent communication.

The associate editor coordinating the review of this manuscript and approving it for publication was Anandakumar Haldorai¹.

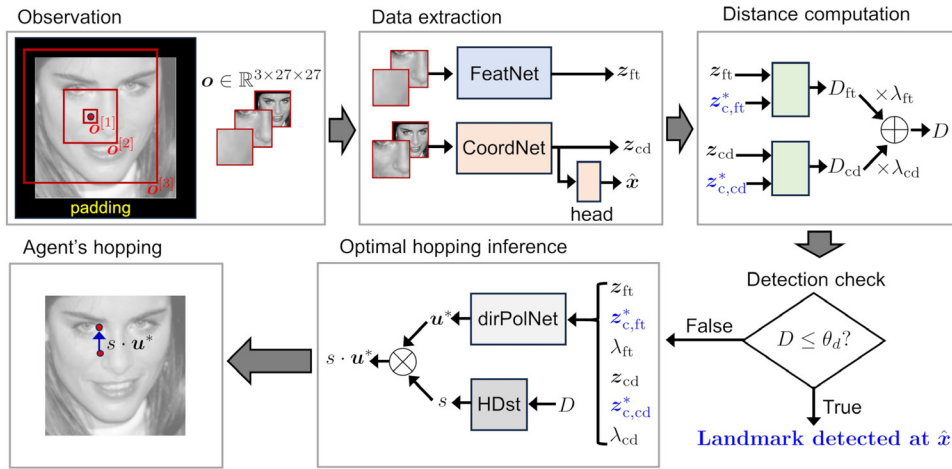


FIGURE 1. Overview of our algorithm for facial landmark detection.

- We enable agents to adaptively determine the optimal hopping direction and hopping distance for efficient landmark localization.
- We model prior knowledge in both latent feature and coordinate spaces via class-conditional generative models, enabling robust landmark localization under weak supervision.
- Despite slightly higher NME than SoTA models, our method achieves $16.8\times$ lower space complexity and $41.1\times$ lower time complexity than the best lightweight baseline.

II. RELATED WORK

A. REGRESSION-BASED METHODS

Facial landmark detection methods are commonly categorized into coordinate regression (CR) and heatmap regression (HR) approaches. CR methods directly predict landmark coordinates using deep neural networks, learning both spatial mappings and local features [4], [5], [10]. While conceptually straightforward, CR approaches are highly sensitive to noise and bias, and typically require extensive supervision to achieve accurate predictions. HR methods generate heatmaps for each landmark, from which coordinates are derived via a decoding step [6], [7]. Despite their strong accuracy, HR models face two notable limitations:

- **Quantization error:** Because heatmaps are typically of lower resolution than the input image, the decoding process introduces quantization errors, degrading coordinate precision [11], [12].
- **Lack of landmark correlation modeling:** Standard HR methods generate heatmaps independently for each landmark, providing limited explicit modeling of landmark relationships.

Furthermore, CR and HR methods require feature extraction over the entire image and combine feature extraction with coordinate prediction in a single large model, leading to high parameter counts and computational cost.

B. MULTIPLE LANDMARK DETECTION WITH AGENTS

Detecting multiple landmarks with agents is challenging due to the need for coordination and reliance on partial visual observations. A key difficulty lies in effectively leveraging both morphological and spatial prior knowledge to maintain robust landmark localization. MARL [13] uses a Deep Q-Network with inter-agent communication to implicitly capture morphological relationships through joint actions. The Multiscale Agent [14] method addresses spatial relations by incorporating multiscale search. SGMaRL [15] integrates a statistical shape model [16] to refine landmark positions based on spatial structure.

While these approaches exploit prior knowledge through inter-agent coordination or global landmark structure modeling, their dependence on coordinated landmark relationships can limit robust landmark localization in complex visual conditions. Furthermore, they typically handle morphological and spatial priors separately rather than jointly.

C. CONTRASTIVE LEARNING

Contrastive learning projects representations into a latent space where similar instances are pulled together and dissimilar ones are pushed apart [17], [18]. Typically, two augmented views per sample yield $2N$ views for a batch of N samples. In self-supervised settings, only views from the same source are considered positive pairs. A limitation is that different-class instances may be treated as negatives, even if semantically related. Supervised Contrastive Loss (SupConLoss) [19] addresses this by incorporating label information, allowing multiple positive pairs per class. This improves representation quality, leading to more accurate and robust classification. Proximity-weighted Contrastive Loss (PWConLoss) [9] further extends supervised contrastive learning by incorporating spatial proximity into positive-pair weighting, enabling spatially structured latent representations.

III. METHOD

A. ALGORITHM OVERVIEW

Fig. 1 illustrates the framework at timestep t . Our method employs N_c agents, each assigned to search a specific landmark ($c \in \mathbf{CI}$; $|\mathbf{CI}| = N_c$) on a given image of size $C \times H \times W$. In total, the agents simultaneously search all N_c landmarks, with each agent exclusively responsible for one landmark. Their coordinates $\mathbf{x} = (x_h, x_w)$ correspond to multiple fixation points, which are normalized to the image size ($x_h, x_w \in [-1, 1]$). We also use normalized coordinate for landmarks. Agents have access to:

- 1) Prior knowledge of landmarks in both the coordinate and latent feature spaces.
- 2) Multiscale local observations $\mathbf{o} \in \mathbb{R}^{3C \times a \times a}$, with $a \ll H, W$.

Agents **do not** know their absolute positions; positions are inferred from observations $\mathbf{o}$. No inter-agent communication occurs.

1) OBSERVATION BY AGENTS

To let each agent understand how the local visual structure relates to its spatial context, we consider three patches ($o^{[1]}: C \times a \times a$, $o^{[2]}: C \times s_1 a \times s_1 a$, and $o^{[3]}: C \times s_2 a \times s_2 a$; $1 < s_1 < s_2$) centered at its current location $\mathbf{x}$. The $o^{[2]}$ and $o^{[3]}$ patches are resized to $C \times a \times a$ and concatenated with the $o^{[1]}$ patch to construct the observation $\mathbf{o} \in \mathbb{R}^{3C \times a \times a}$. Throughout this study, we set $a = 27$, $s_1 = 4$, and $s_2 = 10$, respectively. When the patches exceeds the original image boundaries, constant padding is applied when patches exceed image boundaries.

2) DATA EXTRACTION FROM LOCAL OBSERVATION

Observation $\mathbf{o}$ by an agent located at $\mathbf{x}$ is processed to extract the latent feature and coordinate of the local view in its spatial context using the following models.

- FeatNet: $\mathbb{R}^{2C \times a \times a} \rightarrow \mathbb{R}^{d_{\text{ft}}}$. FeatNet projects current local observation $\mathbf{o}$ by an agent at $\mathbf{x}$ to an embedding $\mathbf{z}_{\text{ft}} \in \mathbb{R}^{d_{\text{ft}}}$. This network projects observation $\mathbf{o}$ to an embedding similar to spatially proximal landmarks. Tensor $\mathbf{o}_{0:2C,:,:}$ is provided as input. $\mathbf{z}_{\text{ft}} = \text{FeatNet}(\mathbf{o}_{0:2C,:,:})$. We omit $\mathbf{o}_{2C:3C,:,:}$ due to its minimal contribution. Unless otherwise stated, we fix $d_{\text{ft}} = 128$.
- CoordNet: $\mathbb{R}^{2C \times a \times a} \rightarrow \mathbb{R}^{2+d_{\text{cd}}}$, which infers the agent's current absolute coordinate. CoordNet function infers the agent's current coordinate $\hat{\mathbf{x}}$ from the local observation. Simultaneously, CoordNet projects the observation to a d_{cd} -dimensional vector $\mathbf{z}_{\text{cd}} (\equiv \hat{\mathbf{x}})$. We omit $\mathbf{o}_{0:C,:,:}$ due to its minimal contribution. $[\hat{\mathbf{x}}, \mathbf{z}_{\text{cd}}] = \text{CoordNet}(\mathbf{o}_{C:3C,:,:})$. We fix $d_{\text{cd}} = 128$.
- RelCoordNet: $\mathbb{R}^{2(C+1) \times a \times a} \rightarrow \mathbb{R}^{2+d_{\text{cd}}}$. This function infers the agent's current relative coordinate ($\Delta\hat{\mathbf{x}}; \Delta\hat{x}_h, \Delta\hat{x}_w \in [-2, 2]$) with reference to a given coordinate $\mathbf{x}^0$. We used RelCoordNet instead of CoordNet for landmarks with high positional variability.

Algorithm 1 Generative Model for Prior Knowledge

Input: Training dataset $\mathcal{T}$ of $N_{\mathcal{T}}$ samples.

Output: Generative model parameters $\mu_{\mathbf{z}_{\text{ft}}|c}$, $\sigma_{\mathbf{z}_{\text{ft}}|c}^2$, $\mu_{\mathbf{z}_{\text{cd}}|c}$, $\sigma_{\mathbf{z}_{\text{cd}}|c}^2$

- 1: $\mathbf{K}_{\text{ft}} \leftarrow$ zero tensor of shape $(N_{\mathcal{T}}, N_c, d_{\text{ft}})$
- 2: $\mathbf{K}_{\text{cd}} \leftarrow$ zero tensor of shape $(N_{\mathcal{T}}, N_c, d_{\text{cd}})$
- 3: **for** each sample y_i in $\mathcal{T}$ **do**
- 4: **for** each landmark c_j in $\mathbf{C}$ **do**
- 5: Compute $\mathbf{o}$ for landmark c_j in sample y_i
- 6: $\mathbf{K}_{\text{ft}}[i, j] \leftarrow \text{FeatNet}(\mathbf{o}_{0:2C,:,:})$
- 7: $\mathbf{K}_{\text{cd}}[i, j] \leftarrow \text{CoordNet}(\mathbf{o}_{C:3C,:,:})$
- 8: **end for**
- 9: **end for**
- 10: $\mu_{\mathbf{z}_{(\cdot)}|c_j} \leftarrow \mathbf{K}_{(\cdot)}.\text{mean}(\text{dim} = 0)$ for $(\cdot) \in \{\text{ft}, \text{cd}\}$
- 11: $\sigma_{\mathbf{z}_{(\cdot)}|c_j}^2 \leftarrow \mathbf{K}_{(\cdot)}.\text{var}(\text{dim} = 0)$ for $(\cdot) \in \{\text{ft}, \text{cd}\}$

3) COMPUTATION OF DEVIATION FROM KNOWLEDGE

For current observation $\mathbf{o}$, the latent feature and coordinate embeddings ($\mathbf{z}_{\text{ft}}$ and $\mathbf{z}_{\text{cd}}$) are each compared to their respective preferred embeddings (prior knowledge $\mathbf{z}_{\text{c,ft}}^* \in \mathbb{R}^{d_{\text{ft}}}$ and $\mathbf{z}_{\text{c,cd}}^* \in \mathbb{R}^{d_{\text{cd}}}$ for landmarks in class c) to compute the distance D .

$$D = \lambda_{\text{ft}} D_{\text{ft}} + \lambda_{\text{cd}} D_{\text{cd}},$$

$$D_{(\cdot)} = \|\mathbf{z}_{(\cdot)} - \mathbf{z}_{\text{c,}(\cdot)}^*\|_2^2, \quad \text{where } (\cdot) \in \{\text{ft}, \text{cd}\}. \quad (1)$$

The distance D is a weighted combination of distances from each embedding space, using balance parameters (λ_{ft} and λ_{cd}) such that $\lambda_{\text{ft}} + \lambda_{\text{cd}} = 1$. If the distance D is lower than a preset threshold θ_d , the agent has successfully arrived at the landmark, outputting its current coordinate ($\hat{\mathbf{x}}$ if CoordNet was used, and $\mathbf{x}^0 + \Delta\hat{\mathbf{x}}$ if RelCoordNet was used). Otherwise, the agent continues searching.

4) HOPPING POLICY

The agent infers optimal hopping direction and distance using PolNet that is factorized into two sub-functions (dirPolNet and HDst): $\text{PolNet} = \text{HDst} \cdot \text{dirPolNet}$.

- dirPolNet: $\mathbb{R}^{2d_{\text{ft}}+2d_{\text{cd}}+2} \rightarrow \mathbb{R}^8$. dirPolNet is a categorical classifier that infers the optimal hopping direction $\mathbf{u}^* \in \mathbf{U}$. We defined the direction space $\mathbf{U}$ as follows: $\mathbf{U} = \{(u_1, u_2) \mid (u_1 \in \mathbf{U}_0 \vee u_2 \in \mathbf{U}_0) \wedge (u_1, u_2 \neq 0)\}$, where $\mathbf{U}_0 = \{-1, 0, 1\}$. Therefore, $|\mathbf{U}| = 8$. This model takes $\mathbf{z}_{\text{ft}}/\mathbf{z}_{\text{c,ft}}^*/\lambda_{\text{ft}}$ and $\mathbf{z}_{\text{cd}}/\mathbf{z}_{\text{c,cd}}^*/\lambda_{\text{cd}}$ for the current observation $\mathbf{o}$ as input.
- HDst: $\mathbb{R} \rightarrow \mathbb{Z}_{>0}$. This function determines the hopping distance $s \in [1, s_{\text{max}}]$ as a function of the distance D .

B. PRIOR KNOWLEDGE MODELING

We built generative models for prior knowledge of landmarks projected to two independent spaces (latent feature space $\mathbb{R}^{d_{\text{ft}}}$ and coordinate space $\mathbb{R}^{d_{\text{cd}}}$).

$$p(\mathbf{z}_{\text{ft}}, c) = p(\mathbf{z}_{\text{ft}}|c)p(c) \quad \text{for latent features,}$$

$$p(\mathbf{z}_{\text{cd}}, c) = p(\mathbf{z}_{\text{cd}}|c)p(c) \quad \text{for coordinate,}$$

where c denotes a landmark class. We modeled the class-conditional probability distribution function $p(z|c)$ using a Gaussian function:

$$p(z|c) = \mathcal{N}(\mu_{z|c}, \Sigma_{z|c}) \approx \mathcal{N}(\mu_{z|c}, \sigma_{z|c}^2 \mathbf{I}), \quad (2)$$

where we apply a diagonal approximation to the covariance matrix $\Sigma_{z|c}$ for computational simplicity. The parameters $\mu_{z|c}$ and $\sigma_{z|c}^2$ were separately computed for each of the two embeddings (to z_{ft} and z_{cd}) over all landmarks of the same class in a given training dataset. This procedure is detailed in **Algorithm 1**.

C. NETWORK MODELS

1) FeatNet

This model (2C9(6C9)-9C16-16C32-32C64-FC256-FC128-L2Norm for gray-scale (RGB) images) projects the local observation $\mathbf{o}_{0:2C,:}$ by an agent at $\mathbf{x}$ to the feature space $\mathbb{R}^{d_{ft}}$. We trained FeatNet using proximity-weighted contrastive learning (PWConLoss) [9], which constructs a spatio-semantic latent space that promotes intra-class clustering and inter-class separation while preserving spatial proximity among landmarks. All landmarks within a $o^{[2]}$ patch of $C \times s_1 a \times s_2 a$ size centered at given $\mathbf{x}$ are considered as positive landmarks while the others as negative ones. Furthermore, the degree of positiveness for positive landmarks is weighted according to their distances from $\mathbf{x}$.

For proximity-weighted contrastive learning, the i th sample in a given batch $\mathcal{B}$ includes a single anchor at $\mathbf{x}_A^{(i)}$, which is placed on a landmark $c_{A(i)}$ (at $\mathbf{x}_{c_{A(i)}}^{(i)}$) that is randomly sampled from total N_c landmarks ($\mathbf{x}_A^{(i)} = \mathbf{x}_{c_{A(i)}}^{(i)}$). We define an augmented batch $\mathcal{B}'$ of the same samples (and sequence) as $\mathcal{B}$ but with a random anchor for each sample, and thus unnecessarily $\mathbf{x}_A^{(i)} = \mathbf{x}_{c_{A(i)}}^{(i)}$. We employ the PWConLoss formulation for the batches $\mathcal{B}$ and $\mathcal{B}'$:

$$\begin{aligned} \mathcal{L}^{\text{PWS}} = & \frac{-1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left[\underbrace{\frac{1}{N^{(i)}} \sum_{j \in \mathcal{B} \setminus \{i\}} w^{(i,j)} \mathbb{1}_{\{c_{A(i)} \in \mathcal{P}^{(j)}\}} l^{(i,j)}}_{\text{between } c_{A(i)} \text{ and } c_{A(j)} (j \neq i)} \right. \\ & \left. + \underbrace{\frac{1}{N'^{(i)}} \sum_{j \in \mathcal{B}'} w^{(i,j)} \mathbb{1}_{\{c_{A(i)} \in \mathcal{P}^{(j)}\}} l^{(i,j)}}_{\text{between } c_{A(i)} \text{ and random observations } \mathbf{o}} \right], \\ N^{(i)} = & \sum_{j \in \mathcal{B} \setminus \{i\}} \mathbb{1}_{\{c_{A(i)} \in \mathcal{P}^{(j)}\}}, \quad N'^{(i)} = \sum_{j \in \mathcal{B}'} \mathbb{1}_{\{c_{A(i)} \in \mathcal{P}^{(j)}\}}, \\ l^{(i,j)} = & \log \frac{\exp(z_A^{(i)} \cdot z_A^{(j)} / \tau)}{\sum_{k \in \mathcal{B}_s \setminus \{i\}} \exp(z_A^{(i)} \cdot z_A^{(k)} / \tau)}, \end{aligned} \quad (3)$$

where $\mathcal{P}^{(j)} = \{c \in \mathcal{C} | \mathbf{x}_c^{(j)} \in o^{[2]} \text{ for } \mathbf{x}_A^{(j)}\}$. The weight $w^{(i,j)}$ is given by

$$w^{(i,j)} = 1 + \exp(-0.025 d^{(i,j)}), \quad (4)$$

where $d^{(i,j)}$ denotes the distance between $\mathbf{x}_{c_{A(i)}}^{(j)}$ and $\mathbf{x}_A^{(j)}$. Note that $\mathbf{x}_{c_{A(i)}}^{(j)}$ means the coordinate of the anchor landmark type of the i th sample $c_{A(i)}$ on the j th sample. For $j \in \mathcal{B} \setminus \{i\}$,

the equality $\mathbf{x}_A^{(j)} = \mathbf{x}_{c_{A(i)}}^{(j)}$ holds. In (3), $\mathcal{B}_s = \text{concat}(\mathcal{B}, \mathcal{B}')$, and $z_{c,ft}^{(i)}$ and τ denote the embedding of the anchor in the i th sample and temperature, respectively. The weight $w^{(i,j)}$ in (4) is constrained to $(1, 2]$. The constant 0.025 is chosen so that $w^{(i,j)} = 1.5$ when $d^{(i,j)} = 27$, matching the height/width of the smallest patch $o^{[1]}$ of size $C \times 27 \times 27$.

2) CoordNet/RelCoordNet

CoordNet projects the local observation $\mathbf{o}_{C:3C,:}$ by an agent at $\mathbf{x}$ to the coordinate space $\mathbb{R}^{d_{cd}}$. It consists of four convolutional layers and one linear layer: 2C9(6C9)-9C16-16C32-32C64-FC128-L2Norm for gray-scale (RGB) images. We deploy an additional head for coordinate regression, $\text{Linear}(128 \rightarrow 2) + \text{Tanh}$, which infers the normalized coordinate $\hat{\mathbf{x}}$ ($\hat{x}_h, \hat{x}_w \in [-1, 1]$) for the agent at $\mathbf{x}$. Similar to FeatNet, the main network (except the head) is trained using the PWConLoss in (3) with a weight function $w^{(i,j)} = 2 - 9.26 \cdot 10^{-3} d^{(i,j)}$ instead of (4). We adopt a linear weight since the weight needs to linearly scale with the distance, clipped to $[1, 2]$. We used the constant $9.26 \cdot 10^{-3}$ to set $w^{(i,j)} = 1$ at $d^{(i,j)} = 108$, corresponding to the width/height of the smallest patch $o^{[2]}$ of size $C \times 108 \times 108$ in CoordNet/RelCoordNet. The anchor for each samples in batch $\mathcal{B}$ is placed on a random coordinate on the sample unlike FeatNet (for which the anchor is on a landmark only), and thus, the embedding z_{cd} for a given observation becomes similar to other observations of spatial proximity. This configuration enables latent space to primarily encode 2D spatial relationships rather than landmark-specific semantic identity, allowing the agent to infer the spatial context of the current observation. The additional head is trained using a mean squared error loss function (MSELoss).

RelCoordNet has the same architecture as CoordNet except the first convolutional layer and head for coordinate regression: 4C9-9C16-16C32-32C64-FC128-L2Norm-FC2-2Tanh for gray-scale images. RelCoordNet takes the reference coordinate $\mathbf{x}^0$ (represented using $x_w^0 \mathbf{1}_{1 \times a \times a}$ and $x_h^0 \mathbf{1}_{1 \times a \times a}$; $x_w^0, x_h^0 \in [-1, 1]$) as its input alongside the local observation $\mathbf{o}_{C:3C,:}$, so that the input consists of $2C + 2$ channels. Instead of Tanh, 2Tanh is applied because of the range of relative coordinate $\Delta \hat{\mathbf{x}}$ ($\Delta \hat{x}_h, \Delta \hat{x}_w \in [-2, 2]$). This model is also trained PWConLoss in (3) with random anchors for the samples in batch $\mathcal{B}$ as for CoordNet. The head is trained using MSELoss.

3) PolNet

PolNet infers the optimal hopping direction $\mathbf{u}^*$ and distance $s \in [1, s_{\max}]$ for the current local observation $\mathbf{o}$ using its sub-functions, dirPolNet and HDst, respectively.

$$\text{PolNet} = \text{HDst}(D) \cdot \text{dirPolNet}(I),$$

$$I = \text{concat}(z_{ft}, z_{c,ft}^*, \lambda_{ft}, z_{cd}, z_{c,cd}^*, \lambda_{cd}),$$

$$\text{HDst}(D) = \min((\lceil D / \Delta_D \rceil), s_{\max}), \quad (5)$$

where Δ_D denotes a unit step for uniform quantization of distance D in (1). dirPolNet (FC512-FC256-FC8-Softmax)

Algorithm 2 Delayed Decision Algorithm

Input: $\hat{\Lambda}$, D_{ft} , D_{cd} , θ_d , λ_{ft} , $\hat{x}$
Output: Updated SHT and $\hat{x}_c$

```

1:  $D_{min} \leftarrow \text{MAX}$ ;  $\hat{x}_c \leftarrow \text{NULL}$ 
2: for  $i = 0$  to  $N_\lambda - 1$  do
3:    $D_{tmp} \leftarrow \hat{\Lambda}[i]D_{ft} + (1 - \hat{\Lambda}[i])D_{cd}$ 
4:   if  $\text{SHT}[i, 0] > D_{tmp}$  then
5:      $\text{SHT}[i, 0] \leftarrow D_{tmp}$ ;  $\text{SHT}[i, 1] \leftarrow \hat{x}$ 
6:   end if
7: end for
8:  $i \leftarrow 0$ 
9: while  $\hat{\Lambda}[i] \geq \lambda_{ft}$  do
10:  if  $\text{SHT}[i, 0] \leq \theta_d$  and  $\text{SHT}[i, 0] \leq D_{min}$  then
11:     $\hat{x}_c \leftarrow \hat{x}$ ;  $D_{min} \leftarrow \text{SHT}[i, 0]$ 
12:  end if
13:   $i \leftarrow i + 1$ 
14: end while

```

infers the optimal hopping direction $u^* \in U$, which is trained using supervised learning on a dataset $\mathcal{T}$. We define $U = \{(u_1, u_2) \mid (u_1 \in U_0 \vee u_2 \in U_0) \wedge (u_1, u_2 \neq 0)\}$, where $U_0 = \{-1, 0, 1\}$. That is, $|U| = 8$. For a given image, a pair of coordinate x and landmark $c \in \mathcal{CL}$ are randomly sampled. Similar to the habitual network [20], each sample $y_i = (I_i, \hat{u}_i)$ in $\mathcal{T}$ consists of (a) input I_i (in (5)) for the local observation o at the random coordinate x and (b) $\hat{u}_i = \arg \min_{u \in U} D$ for the coordinate x and landmark c .

D. HYPERPARAMETER SETTING**1) DETECTION THRESHOLD**

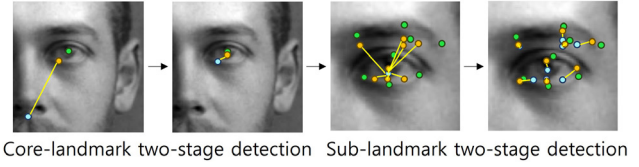
Detection threshold θ_d is a primary hyperparameter that determines the detection accuracy and speed, which are measured in NME and the number of detection steps (T_d), respectively. T_d denotes the average number of agent hopping steps required to detect landmarks in a single image. A lower θ_d generally yields a lower NME but a larger T_d . To balance this trade-off, threshold θ_d is initially set to its minimum ($\theta_{d,min}$) and is monotonously increased by $\Delta\theta_d$ at each timestep if detection fails.

2) BALANCE PARAMETERS

Balance parameters λ_{ft} and $\lambda_{cd}(= 1 - \lambda_{ft})$ in (1) govern the complementary contributions of the distances from distinct representation spaces (D_{ft} and D_{cd}) to the total distance D . A higher λ_{ft} generally yields a lower NME but a larger T_d . We initially set λ_{ft} is initially set to its maximum ($\lambda_{ft,max}$) and monotonically decrease by $\Delta\lambda_{ft}$ once every two timesteps if detection fails.

E. EFFICIENCY ENHANCEMENTS**1) DELAYED DECISION ALGORITHM**

Critical detection inefficiency with varying hyperparameters (θ_d and λ_{ft}) arises when previously visited locations with previous hyperparameter values yield successful detection

**FIGURE 2.** Example of cascaded detection.

with the current θ_d . Revisiting such locations and repeatedly recalculating the distance D leads to redundant computation. To mitigate this issue, we introduce a delayed decision algorithm for the following case.

$$\theta_d[t'] < D[t'] = \sum_{i \in \{ft, cd\}} \lambda_i[t'] D_i[t'] \leq \theta_d[t] \text{ for } t' < t.$$

We define a balance parameter set $\Lambda = \{\lambda[t] \mid \forall t \in [0, t_{max}]\}$ ($|\Lambda| = N_\lambda$) and a corresponding tuple $\hat{\Lambda}$ of $\lambda(\in \Lambda)$ sorted in descending order. This delayed decision algorithm is based on a $N_\lambda \times 2$ search history table (SHT). **Algorithm 2** explains SHT organization and delayed decision based on the SHT.

2) TWO-STAGE DETECTION

To reduce the number of detection steps T_d , we introduce a two-stage detection strategy for all landmarks. The first stage detects coarse coordinates of landmarks using the higher detection threshold $\theta_d^{(1)}$ and lower balance parameter $\lambda_{ft}^{(1)}$, and the second stage refines the coordinate using the lower threshold $\theta_d^{(2)} (< \theta_d^{(1)})$ and higher balance parameter $\lambda_{ft}^{(2)} (> \lambda_{ft}^{(1)})$. In each stage, the hyperparameters change following the rule explained in the previous section.

3) CASCADED DETECTION

Facial landmarks can be grouped based on their spatial proximity in the latent feature space $\mathbb{R}^{d_n}$. We choose a single core-landmark for each group and considered the others in the same group as sub-landmarks. Agents responsible for detecting landmarks in the same group often exhibit overlapping trajectories in the initial detection phase, leading to redundant computation. To address this, we propose a cascaded detection strategy. In this strategy, a single agent first detects the core-landmark using FeatNet and CoordNet for local observation o at each timestep t . This core-landmark serves as a reference. Subsequently, multiple agents simultaneously search the sub-landmarks with reference to the core-landmark. This approach mitigates redundant agent movements during the early detection steps, thereby reducing redundant computation and processing time. Fig. 2 shows an example of cascaded detection for a Right-eye group.

4) PER-STAGE TIMEOUT

To prevent unbounded roaming of agents, we impose a per-stage timeout (30 steps). If the distance D does not fall below the detection threshold θ_d within the period, the agent terminates hopping and its position is finalized by the delayed-decision algorithm.

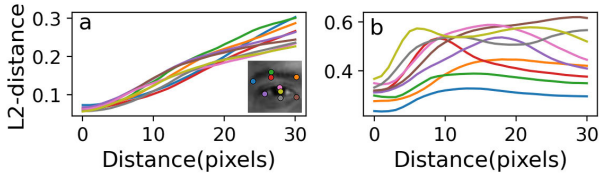


FIGURE 3. L2 distances between feature embeddings and landmarks at varying spatial distances from the landmarks in the inset for FeatNet (a) with PWConLoss and (b) with supervised contrastive learning baseline.

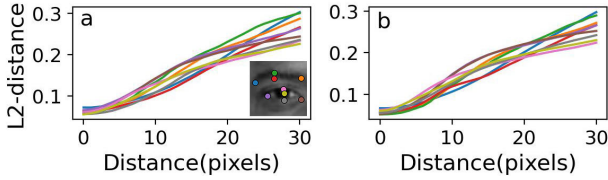


FIGURE 4. L2 distances between feature embeddings and landmarks at varying spatial distances from the landmarks in the inset for FeatNet (a) with $o_{0:2C,,:}$ and (b) with o .

IV. EXPERIMENTAL RESULTS

We used the COFW [21] and 300W [22] datasets. COFW comprises 1,345 training and 507 test images (gray-scale), each annotated with 29 landmarks. COFW with frequent occlusions is well-suited for evaluating our method's ability to leverage prior knowledge of landmarks. 300W comprises 3,148 training and 689 test images (RGB), each annotated with 68 landmarks. This dataset exhibits a wide range of variations in pose and lighting conditions.

A. IMPLEMENTATION DETAILS

Each image is cropped to include the full head, resized to 256×256 , then randomly rescaled ($\pm 5\%$) and horizontally flipped (50%). For cascaded detection, the landmarks in each image are grouped and their core-landmarks are selected based on their spatial proximity in the latent feature space, their detectability, and computational efficiency. Landmarks that are close to each other in the latent feature space are assigned to the same group. For each group, the core-landmark was selected to have distinct local visual characteristics, allowing it to be detected more reliably than neighboring sub-landmarks and to serve as a stable reference point for relative localization. In addition, each group was disallowed to include more than 20 landmarks to avoid excessive sub-landmark search. Based on these criteria, the landmark groups and their core-landmarks were defined as follows.

COFW: left_eye (left pupil), right_eye (right pupil), and others (nose tip)

300W: left_eye (left inner canthus), right_eye (right inner canthus), mouth (Cupid's bow), nose (nose tip), jaw_line (none).

For the jaw_line group in the 300W dataset, no core-landmark was assigned because the jaw-line does not contain a visually distinct landmark. We used different FeatNets with

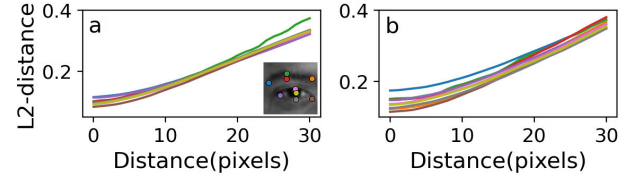


FIGURE 5. L2 distances between coordinate embeddings and landmarks at varying spatial distances from the landmarks in the inset for (a) CoordNet and (b) RelCoordNet.

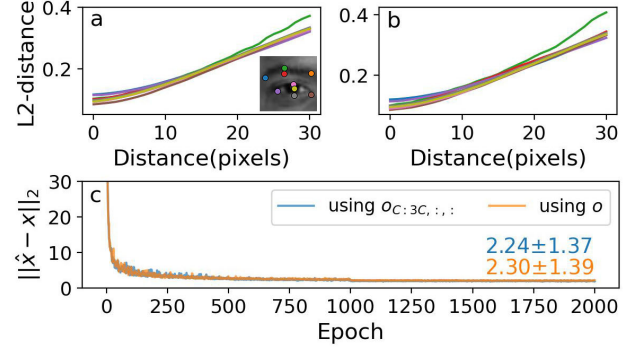


FIGURE 6. L2 distances between coordinate embeddings and landmarks at varying spatial distances from the landmarks in the inset for CoordNet with (a) $o_{C:3C,,:}$ and (b) o . (c) Coordinate regression error for CoordNet with $o_{C:3C,,:}$ and o .

different sets of parameters (but the same CoordNet and RelCoordNet) for each group. For the Others group in COFW, RelCoordNet was used for detecting the sub-landmarks. Note that we used the means $\mu_{z_{ft|c}}$ and $\mu_{z_{cd|c}}$ as prior knowledge $z_{c,ft}^*$ and $z_{c,ft}^*$, respectively, unless otherwise stated. The models were trained using the Pytorch framework [23] on a GPU workstation (RTX A6000; Xeon Gold CPU 2.9GHz; 256 GB DRAM). Landmark detection experiments were conducted on both the GPU workstation and a desktop equipped with an i5 CPU (2.5GHz) and 32 GB DRAM. The hyperparameters were optimized using Optuna [24].

B. PERFORMANCE OF NETWORK MODELS

1) FeatNet

Fig. 3 shows the L2 distance between the feature embedding z_{ft} and several landmarks (in the inset) with spatial distance. Fig. 3a highlights (a) a gradual increase in L2 distance with distance between the observation o and landmark and (b) marginal variability in L2 distance at zero distance upon different landmarks. These properties provide useful spatial cues for adaptive landmark search, enabling the HDst in PolNet to estimate the optimal hopping distance from latent-space distances. As a counterpart, we used the same network trained with supervised contrastive learning, where spatial proximity was addressed in binary form, employing the same loss as in (3) but without the proximity weight $w^{(i,j)}$. The performance of this counterpart is shown in Fig. 3b. Fig. 4 represents the results of using $o_{0:2C,,:}$ and the full

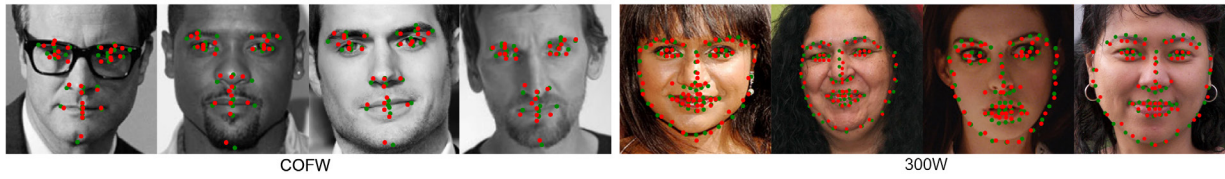


FIGURE 7. Detected landmarks (red circles) and ground-truth annotations (green circles) on sample images from COFW and 300W.

TABLE 1. Comparison of our method with SoTA approaches on COFW and 300W datasets. The NME value in parentheses for 300W excludes the jaw_line landmarks.

Method	COFW		300W	# Params (M)	FLOPs	# Detection Steps T_d
	NME-ocular	NME-pupil	NME-ocular			
LAB [25]	3.92	5.58	3.49	25.1	18.9G	-
AWing [26]	-	4.94	3.07	24.2	26.8G	-
AVS [10]	-	4.43	3.86	28.3	2.40G	-
HRNet [6]	3.45	-	3.32	9.66	4.75G	-
PIP [27]	-	-	3.36	12.0	2.40G	-
ADNet [7]	-	4.69	2.93	13.4	17.0G	-
SDFL [5]	3.63	-	-	-	5.17G	-
HIH [12]	3.21	4.63	3.09	22.7	17.2G	-
SLPT [28]	3.32	4.63	3.17	13.2	6.12G	-
STARLoss [29]	-	4.62	2.87	13.4	-	-
D-ViT [8]	-	4.13	2.85	67.3	21.8G	-
PoPos [30]	-	3.80	3.28	9.70	1.20G	-
Ours on COFW/300W	8.28 ± 0.08	11.96 ± 0.12	9.36(8.33) ± 0.05	0.577	21.1± 0.8M / 29.1± 1.1M	10.42± 0.36 / 12.77± 0.52

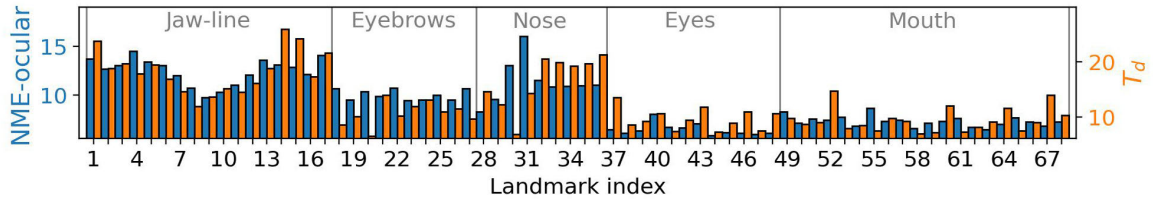


FIGURE 8. Average detection accuracy (NME) and number of detection steps (T_d) for each landmark on 300W.

observation $\mathbf{o}$ as the input to FeatNet. The results indicate that the difference is negligible.

2) CoordNet/RelCoordNet

Fully trained CoordNet and RelCoordNet successfully infer the coordinate of the current observation as $\mathbf{z}_{cd}$ -dimensional embeddings. Fig. 5 shows the L2 distance between the coordinate embedding $\mathbf{z}_{cd}$ and several landmarks (in the inset) with spatial distance. Compared with Fig. 3, CoordNet/RelCoordnet can infer the coordinate of distal observations with higher precision (lower variability). Fig. 6 represents the results of using $\mathbf{o}_{C:3C,,:}$ and the full observation $\mathbf{o}$ as the input to CoordNet. The results indicate that the difference is negligible.

C. DETECTION PRECISION AND EFFICIENCY

Algorithm 3 summarizes the overall procedure of the two-stage detection process for landmark c . The detected landmarks on several samples in COFW and 300W are shown in Fig. 7, demonstrating successful landmark detection using our algorithm. Nevertheless, our method has higher NME

than regression-based SoTA techniques as listed in Table 1. This is largely due to the fact that our approach relies on prior knowledge of landmark features rather than direct supervised learning, while the current prior knowledge model has limited expressiveness. For instance, although our method well detects the jaw-line landmarks in 300W samples (Fig. 7), NME for these landmarks is large due to their deviation from the semi-automatically annotated jaw-line landmarks on 300W samples. Fig. 8 plots the average NME and the number of detection steps T_d for each landmark on 300W sample, identifying high NME for the jaw_line landmarks.

However, our method demonstrates significantly low computational complexity.

- **Space complexity:** Total 577k parameters (FeatNet/CoordNet/PolNet with 123k/58k/396k parameters).
- **Time complexity:** Total 21.1 MFLOPs for a 10.42 T_d on COFW and 29.1 MFLOPs for a 12.77 T_d on 300W.

Relative to the lightweight PoPos [30] model with 9.70M parameters and 1.20G FLOPs, our method reduces space complexity by 16.8 $\times$ and time complexity by 41.1 $\times$.

Input: c , Image, $\mathbf{x}$, $t_{\max}$, $\boldsymbol{\theta}_{\text{d,min}} \in \mathbb{R}^2$, $\Delta\boldsymbol{\theta}_{\text{d}} \in \mathbb{R}^2$, $T_{\text{d,up}} \in \mathbb{Z}_{>0}^2$, $\boldsymbol{\lambda}_{\text{ft,max}} \in \mathbb{R}^2$, $\Delta\boldsymbol{\lambda}_{\text{ft}} \in \mathbb{R}^2$, $T_{\text{d,down}} \in \mathbb{Z}_{>0}^2$
Output: $\hat{\mathbf{x}}_c$

```

1:  $z_{c,ft}^*, z_{c,cd}^* \leftarrow$  Prior knowledge modeling( $c$ )
2:  $\hat{x}_c \leftarrow$  NULL
3: for stage = 0 to 1 do
4:    $\theta_d \leftarrow \theta_{d,min}[stage]$ 
5:    $\Delta\theta_d \leftarrow \Delta\theta_d[stage]$ 
6:    $T_{d,up} \leftarrow T_{d,up}[stage]$ 
7:    $\lambda_{ft} \leftarrow \lambda_{ft,max}[stage]$ 
8:    $\lambda_{cd} \leftarrow 1 - \lambda_{ft}$ 
9:    $\Delta\lambda_{ft} \leftarrow \Delta\lambda_{ft}[stage]$ 
10:   $T_{d,down} \leftarrow T_{d,down}[stage]$ 
11:   $\hat{\Lambda} \leftarrow \{\lambda_{ft}[t] | \forall t \in [0, t_{max}]\}$ 
12:  for  $t = 0$  to  $t_{max}$  do
13:     $o \leftarrow$  Observation by agent(Image,  $x$ )
14:     $z_{ft}, z_{cd}, \hat{x} \leftarrow$  Data extraction( $o$ )
15:     $D_{ft}, D_{cd}, D \leftarrow$  DistComp( $z_{(\cdot)}, z_{c,(\cdot)}^*, \lambda_{(\cdot)}$ ),
      where  $(\cdot) \in \{ft, cd\}$ 
16:     $\hat{x}_c \leftarrow$  Delayed decision( $\hat{\Lambda}, D_{ft}, D_{cd}, \theta_d, \lambda_{ft}, \hat{x}$ )
17:    if  $\hat{x}_c \neq$  NULL then
18:       $x \leftarrow \hat{x}_c$ 
19:      Terminate.
20:    end if
21:    if  $t \geq T_{d,up}$  then
22:       $\theta_d \leftarrow \theta_d + \Delta\theta_d$ 
23:    end if
24:    if  $t \geq T_{d,down}$  and  $t - T_{d,down} = \text{even}$  then
25:       $\lambda_{ft} \leftarrow \lambda_{ft} - \Delta\lambda_{ft}$ 
26:    end if
27:     $s \cdot u^* \leftarrow$  Hopping policy( $D, z_{(\cdot)}, z_{c,(\cdot)}^*, \lambda_{(\cdot)}$ )
      where  $(\cdot) \in \{ft, cd\}$ 
28:     $x \leftarrow x + s \cdot u^*$ 
29:  end for
30: end for

```

TABLE 2. Detection performance on COFW under various degradations.

Degradation mode	NME	T_d	FLOPs
Baseline	8.28	10.42	21.1M
Blur ($\sigma = 1$)	8.64	14.51	29.4M
Blur ($\sigma = 2$)	8.72	14.87	30.1M
Blur ($\sigma = 3$)	8.90	15.72	31.8M
JPEG ($Q = 80$)	8.59	14.47	29.3M
JPEG ($Q = 60$)	8.65	14.42	29.3M
JPEG ($Q = 40$)	8.59	14.43	29.3M
JPEG ($Q = 20$)	8.64	14.46	29.3M
Motion blur ($k = 5$)	8.71	14.62	29.6M
Motion blur ($k = 10$)	8.86	15.29	31.0M
Occlusion (size=(20, 40))	8.74	14.78	29.9M

D. ABLATION STUDY

1) ROBUSTNESS TO SAMPLE DEGRADATIONS

To validate robustness of proposed method, we evaluate the detection performance under various degradation conditions, including Gaussian blur, JPEG degradation, motion blur,

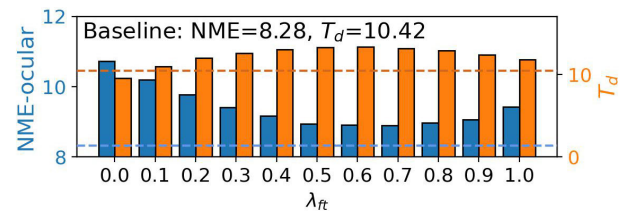


FIGURE 9. Relationship between detection accuracy and number of detection steps for different fixed values of λ_{ft} on COFW.

TABLE 3. Impact of techniques on detection performance.

	Delayed decision	Two-stage detection	Cascaded detection	NME	T_d	FLOPs
Baseline	✓	✓	✓	8.28	10.42	21.1M
Case 1	✗	✓	✓	10.35	14.56	29.5M
Case 2	✓	✗	✓	8.52	7.81	15.8M
Case 3	✓	✓	✗	9.46	20.97	42.4M
Case 4	✗	✗	✓	10.46	9.25	18.7M
Case 5	✗	✓	✗	12.50	30.32	61.4M
Case 6	✓	✗	✗	9.53	16.20	32.8M
Case 7	✗	✗	✗	12.55	33.65	68.1M

and occlusion. The results are summarized in Table 2. Across all degradation types, our method exhibits consistently moderate performance degradation. While blurring with $\sigma = 3$ causes the largest NME loss, it does not exceed 7.5%.

2) BALANCE PARAMETER

The balance parameter λ_{ft} is an important hyperparameter that governs both detection accuracy and number of detection steps, and its value is scheduled over time. We analyzed the impact of λ_{ft} on detection accuracy and number of detection steps by varying its value over the range $[0, 1]$ while keeping it fixed during each detection run. As shown in Fig. 9, λ_{ft} exhibits a clear trade-off between accuracy and number of detection steps, highlighting the need for parameter scheduling to achieve optimal performance.

3) PERFORMANCE ENHANCEMENT TECHNIQUES

To enhance detection performance, we employ three techniques: delayed decision, two-stage detection, and cascaded detection. We analyze the impact of each technique on detection accuracy and number of detection steps. The corresponding results are presented in Table 3.

4) HYPERPARAMETER SCHEDULING

Our method employs hyperparameter scheduling for accurate and efficient detection. The ablation results of hyperparameter scheduling on COFW are provided in Table 4. The number of detection steps increases by $2.69\times$ without θ_d scheduling and by $3.27\times$ without both θ_d and λ_{ft} scheduling.

5) FAILURE CASE ANALYSIS

The hopping policy does not always guarantee convergence. When an agent fails to detect within a predefined

TABLE 4. Detection performance on COFW under ablation of the hyperparameter scheduling.

	λ_{ft} scheduling	θ_d scheduling	NME	T_d	FLOPs
Baseline	✓	✓	8.28	10.42	21.1M
Case 1	✗	✓	8.92	11.40	23.1M
Case 2	✓	✗	8.77	28.00	56.7M
Case 3	✗	✗	8.76	34.04	68.9M

**FIGURE 10.** Trajectories of agents that fail to detect the target landmark of COFW. The blue circle represents the starting point, the yellow circles represent the intermediate trajectory, the red circle represents the final detection point, and the green circle represent the ground-truth landmark, respectively.

number of steps, it invokes the delayed decision mechanism to select the most plausible point as the final landmark. Fig. 10 illustrates the trajectories of agents across stages and timesteps in failure cases. Although occlusion prevents the agent from reaching the true landmark, it ultimately chooses the optimal location based on its prior knowledge using the delayed-decision algorithm with SHT.

V. LIMITATIONS AND FUTURE WORK

In our proof-of-concept, each landmark's prior was modeled as a diagonal Gaussian with a single global mean $\mu_{z(\cdot)}$. This likely over-simplifies landmark variability across samples, resulting in lower detection accuracy than state-of-the-art. We replaced this with a Gaussian mixture model comprising five components, with means initialized from K -means clustering ($K = 5$). Selecting the best component per sample improved accuracy on COFW by approximately 28% as shown in Table 5. As an upper-bound analysis, using per-sample landmark features as priors yields accuracy close to state-of-the-art (From samples in Table 5). These results indicate that (i) accuracy is primarily limited by the knowledge model and (ii) the other sub-modules (FeatNet, CoordNet, PolNet) function as intended. Our results show a strong correlation between prior knowledge and detection accuracy, suggesting that building a more expressive and robust prior is a key direction for future work.

Regarding scalability, the proposed framework is likely extended to datasets with a larger number of landmark classes because the prior knowledge model is defined in a class-conditional manner. Each landmark class is represented by its own prior parameters in the feature and coordinate spaces, while the network sub-modules, such as FeatNet, CoordNet, and PolNet, can be shared across landmarks or landmark groups. Thus, increasing the number of landmark classes mainly results in an approximately linear increase in the prior parameters, rather than requiring a redesign of the overall agent-based detection framework.

TABLE 5. Detection performance on COFW with different prior knowledge models.

	Baseline	Sampling	Gaussian mixture	From samples
NME	8.28	8.92	5.99	3.88
T_d	10.42	20.35	13.41	10.39

However, more complex datasets with many landmark classes may exhibit larger intra-class variations and more ambiguous local appearances, making the design of more expressive yet compact prior models increasingly important. The Gaussian mixture model experiment in Table 5 supports this direction, showing that improving prior expressiveness can enhance detection accuracy while preserving the scalable structure of the proposed framework.

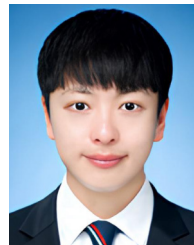
VI. CONCLUSION

We proposed a lightweight, agent-based framework for facial landmark detection that leverages prior knowledge and local observations without relying on strong supervision. Each agent infers its location independently using embeddings from dual spaces—feature and coordinate—guided by class-conditional generative models. We further improved efficiency with a multi-stage detection strategy and delayed decision mechanism to reduce redundant computation. While our method shows slightly higher NME than SoTA approaches, it achieves exceptional efficiency by reducing space complexity by 16.8 $\times$ and time complexity by 41.1 $\times$ compared to the SoTA lightweight model, making it ideal for resource-constrained platforms. Furthermore, experiments with more expressive prior knowledge models suggest that the proposed framework can achieve strong localization accuracy while preserving its computational efficiency. This work demonstrates that prior knowledge-guided agent-based detection is a practical and scalable alternative for efficient landmark localization.

REFERENCES

- [1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv. (CSUR)*, vol. 35, no. 4, pp. 399–458, Dec. 2003.
- [2] H. Yang, U. Ciftci, and L. Yin, "Facial expression recognition by de-expression residue learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2168–2177.
- [3] F. Liu, Q. Zhao, X. Liu, and D. Zeng, "Joint face alignment and 3D face reconstruction with application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 664–678, Mar. 2020.
- [4] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, May 2018, pp. 2235–2245.
- [5] C. Lin, B. Zhu, Q. Wang, R. Liao, C. Qian, J. Lu, and J. Zhou, "Structure-coherent deep feature learning for robust face alignment," *IEEE Trans. Image Process.*, vol. 30, pp. 5313–5326, 2021.
- [6] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.
- [7] Y. Huang, H. Yang, C. Li, J. Kim, and F. Wei, "ADNet: Leveraging error-bias towards normal direction in face alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3080–3090.

- [8] Z. Dang, J. Li, and L. Liu, "Cascaded dual vision transformer for accurate facial landmark detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Feb. 2025, pp. 5884–5894.
- [9] S. Jin and D. Seok Jeong, "WorldComp2D: Spatio-semantic representations of object identity and location from local views," 2026, *arXiv:2605.11743*.
- [10] S. Qian, K. Sun, W. Wu, C. Qian, and J. Jia, "Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10153–10163.
- [11] A. Bulat, E. Sanchez, and G. Tzimiropoulos, "Subpixel heatmap regression for facial landmark localization," 2021, *arXiv:2111.02360*.
- [12] X. Lan, Q. Hu, Q. Chen, J. Xue, and J. Cheng, "HIH: Towards more accurate face alignment via heatmap in heatmap," 2021, *arXiv:2104.03100*.
- [13] A. Vlontzos, A. Alansary, K. Kamnitsas, D. Rueckert, and B. Kainz, "Multiple landmark detection using multi-agent reinforcement learning," in *Proc. 22nd Int. Conf. Med. Image Comput. Comput. Assist. Intervent.*, Shenzhen, China, Cham, Switzerland: Springer, 2019, pp. 262–270.
- [14] A. Alansary, O. Oktay, Y. Li, L. L. Folgoc, B. Hou, G. Vaillant, K. Kamnitsas, A. Vlontzos, B. Glocker, B. Kainz, and D. Rueckert, "Evaluating reinforcement learning agents for anatomical landmark detection," *Med. Image Anal.*, vol. 53, pp. 156–164, Apr. 2019.
- [15] K. Wan, L. Li, D. Jia, S. Gao, W. Qian, Y. Wu, H. Lin, X. Mu, X. Gao, S. Wang, F. Wu, and X. Zhuang, "Multi-target landmark detection with incomplete images via reinforcement learning and shape prior embedding," *Med. Image Anal.*, vol. 89, Oct. 2023, Art. no. 102875.
- [16] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, Jan. 1995.
- [17] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
- [18] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. 16th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 776–794.
- [19] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 18661–18673.
- [20] F. Cushman and A. Morris, "Habitual control of goal selection in humans," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 45, pp. 13817–13822, Nov. 2015.
- [21] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1513–1520.
- [22] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image Vis. Comput.*, vol. 47, pp. 3–18, Mar. 2016.
- [23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8024–8035.
- [24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2623–2631.
- [25] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: A boundary-aware face alignment algorithm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2129–2138.
- [26] X. Wang, L. Bo, and L. Fuxin, "Adaptive wing loss for robust face alignment via heatmap regression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6970–6980.
- [27] H. Jin, S. Liao, and L. Shao, "Pixel-in-pixel Net: Towards efficient facial landmark detection in the wild," *Int. J. Comput. Vis.*, vol. 129, no. 12, pp. 3174–3194, Dec. 2021.
- [28] J. Xia, W. Qu, W. Huang, J. Zhang, X. Wang, and M. Xu, "Sparse local patch transformer for robust face alignment and landmarks inherent relation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4052–4061.
- [29] Z. Zhou, H. Li, H. Liu, N. Wang, G. Yu, and R. Ji, "STAR loss: Reducing semantic ambiguity in facial landmark detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 15475–15484.
- [30] C.-Y. Xiang, J.-Y. He, Z.-Q. Cheng, X. Wu, and X.-S. Hua, "Popos: Improving efficient and robust facial landmark detection with parallel optimal position search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, 2025, pp. 8602–8610.



SEONGMIN JIN received the B.S. degree in materials science and engineering from Hanyang University, Seoul, South Korea, in 2021, where he is currently pursuing the Ph.D. degree in materials science and engineering. His current research interest includes operation-efficient algorithms for image recognition.



DOO SEOK JEONG (Member, IEEE) received the B.E. and M.E. degrees in materials science from Seoul National University, South Korea, in 2002 and 2005, respectively, and the Ph.D. degree in materials science from RWTH Aachen, Germany, in 2008. From 2008 to 2018, he was with Korea Institute of Science and Technology. He is currently a Professor with Hanyang University, South Korea. His research interests include spiking neural networks for sequence learning, future prediction, learning algorithms, spiking neural network design, and digital neuromorphic processor design.

...